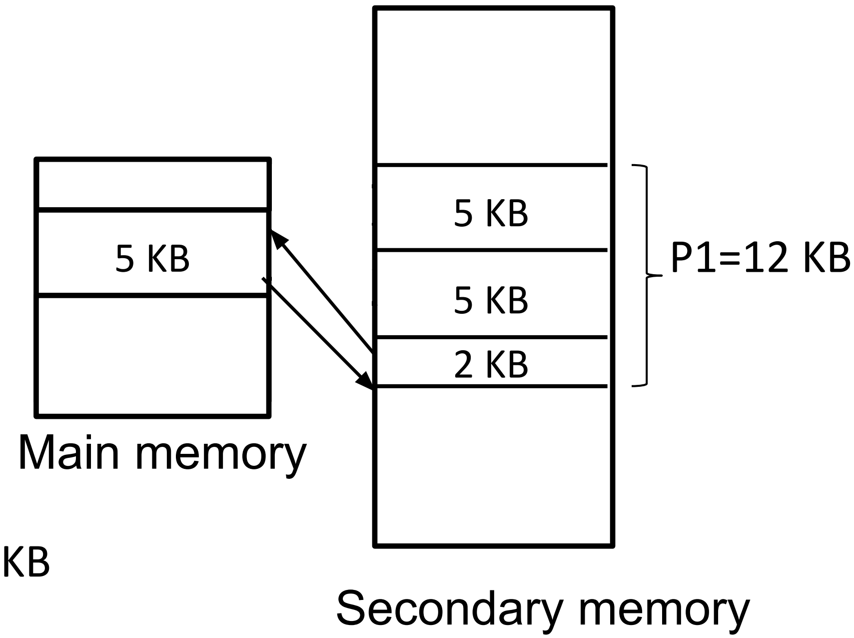
Virtual memory concept



The size of a program P1=12 KB

Available free main memory space = 5 KB

P1 can be divided into three parts 5 KB ,5 KB and 2 KB

1st 5 KB of P1 will be loaded into main memory.

After completion of 1st 5 KB of P1 then 1st portion will be swapped out and next 5 KB will be swapped in to the main memory.

Similarly after completion of next 5 KB of P1 then 2nd portion will be swapped out and next 2 KB will be swapped in to the main memory.

After completion of 2 KB of P1, 2 KB will be swapped out from the main memory.

Virtual memory

- Virtual memory is a technique used in computer system.
- It gives the programmer an illusion of having a large main memory, but it really does not exist.
- The size of the virtual memory is equivalent to the size of secondary memory.
- Each address generated by CPU called virtual address (logical address) is mapped with Physical address in main memory.
- The mapping is performed by MMU (Memory Management Unit).

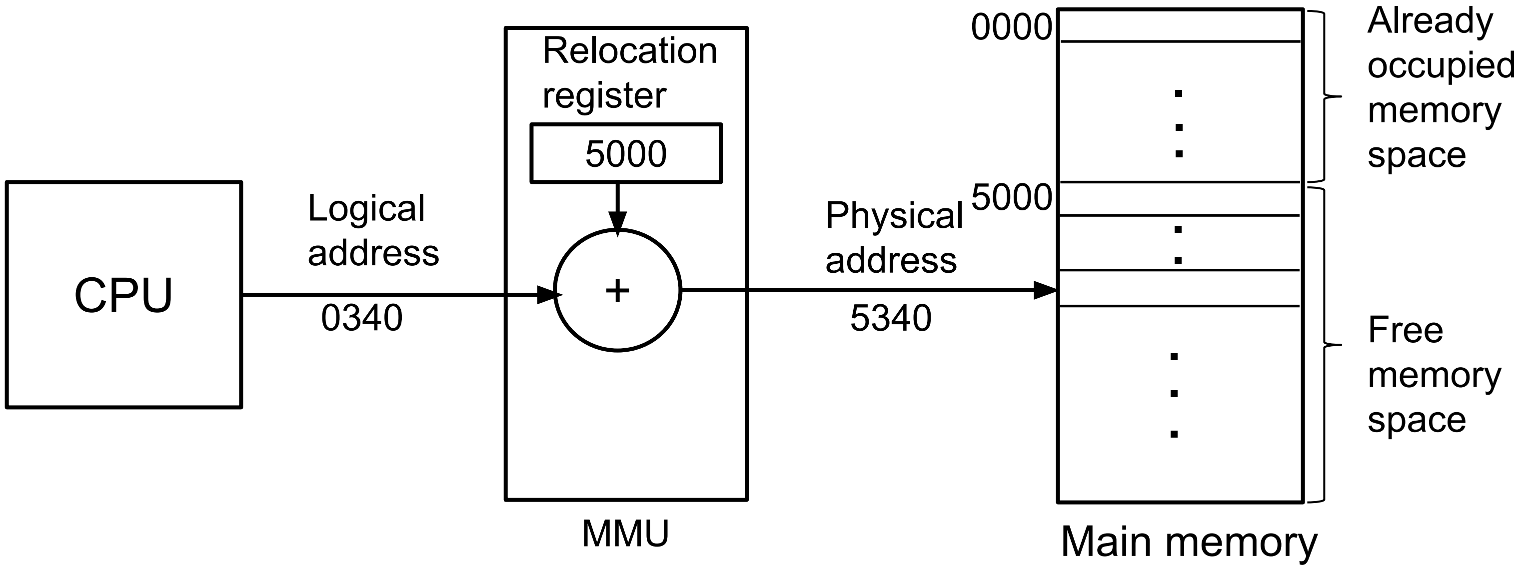
Advantages of virtual memory

i) The program larger than free memory space can be executed by using virtual memory technique.

ii) The programmers do not need to worried about the size of the program.

iii) It allows multiprogramming which increases the CPU utilization.

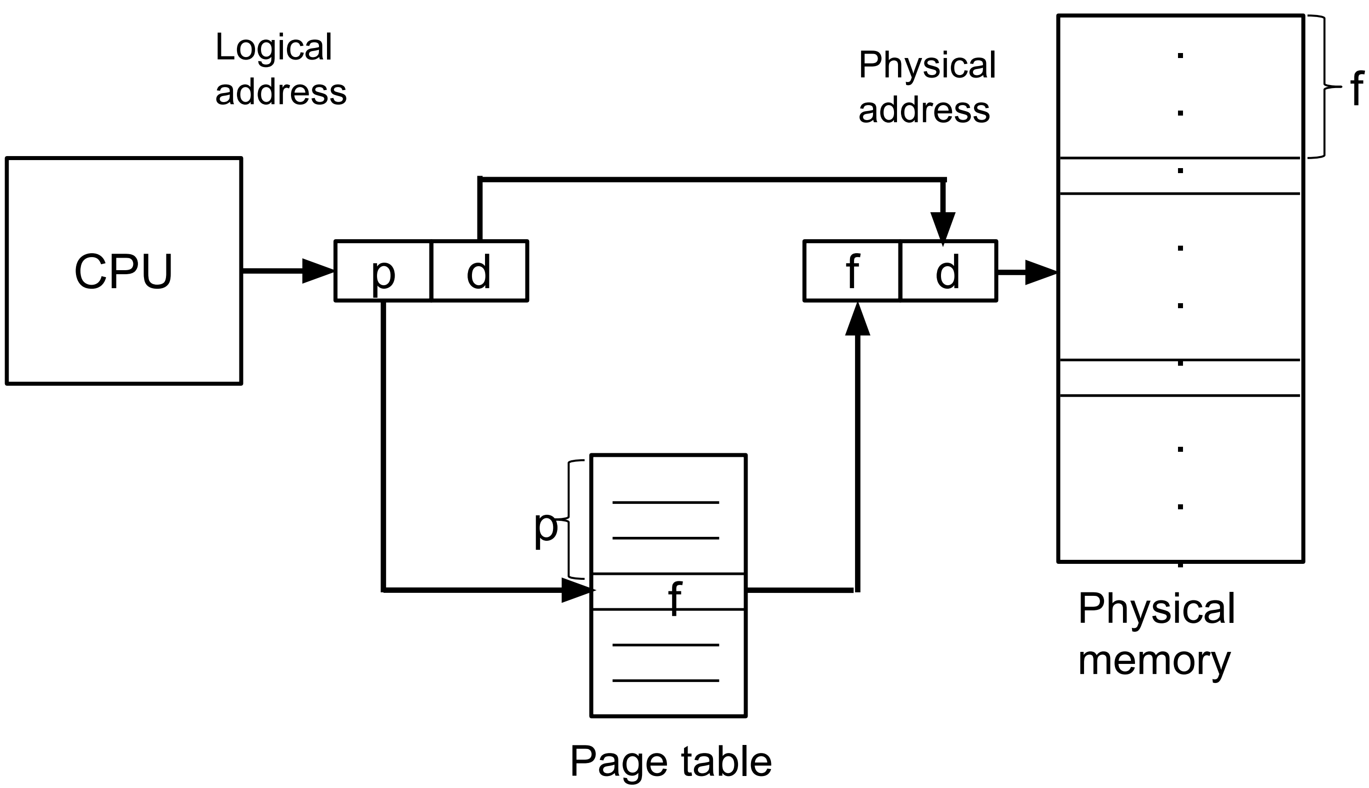
Logical to Physical address translation using MMU



- The addresses generated by CPU or program is called logical addresses.
- The corresponding addresses in the physical memory occupied by the executing program, are called physical addresses.
- The memory management unit (MMU) maps each logical address to physical address

For example, If the relocation register or base register holds an address value 5000. Then logical address 0 will mapped with 5000 address and similarly logical address 0340 is mapped to the physical address 5340.

Paging Hardware



Example of Non-contiguous memory allocation

Example of Non-contiguous memory allocation is Paging.

- The program space will be known as a logical address space and main memory space will be physical address space.
- Total logical address space (generated by the CPU) is divided into equal size partitions and each partition is known as a page.
- Similarly the main memory is also divided into equal size partitions and each partition is known as a frame.
- Page size=frame size.
- So, When a particular page is getting loaded into a certain frame there will be no free space remaining.
- Every logical address is divided into two parts: a page number(p) and a page offset(d).
- The page number is used as an index into page table.
- A page table contains the frame number(f).
- Page table is used to store the information that which pages are mapped with which frames.
- This frame number (f) is combined with the page offset(d) and generate the physical memory address.

Paging contd.

- The size of page is typically a power of 2 which makes the translation of a logical address into a page number and page offset particularly easy.
- The size of logical address space is 2^m and a page size is 2^n bytes .
- The high order m-n bits of logical address represent the page number.
- The lower order n bits of logical address represent the page offset.
- Thus the logical address is as follows:

page number	page offset
p	d
m – n bits	n bits

Where p is an index into the page table and d is the displacement within the page.

Example: The size of logical address space is 64 KB and the page size is 2 KB. How many pages are there? How many bits are used to represent page number and page offset?

To represent logical address 16 bits (2^{16} or 64 KB) are required

Total number of pages = logical address space/ page size= 2^{16} / 2^{11} = 2^5 = 32

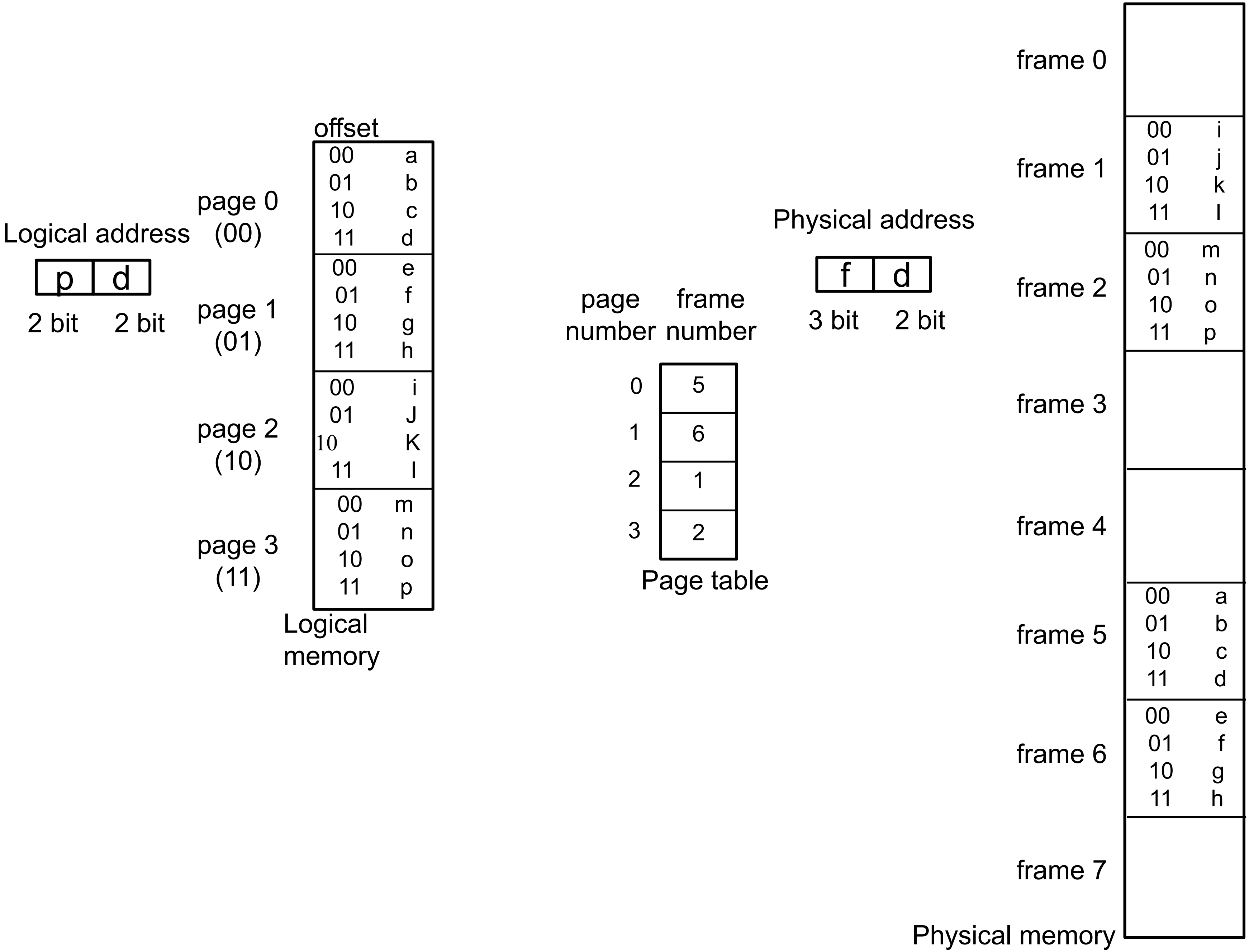
5 bits (16 -11) are representing the page number.

11 bits are representing the page offset.

Paging example for a 32 byte memory with 4 byte pages

Logical address space is 16 bytes (2^4).
Page size is 4 bytes (2^2).
Physical memory of 32 bytes (2^5)
Number of pages = Logical address space /page size = $2^4/ 2^2 = 4$
Since page size = frame size,
number of frames=size of physical memory /frame size = $2^5/ 2^2 = 8$
There are four pages Page0, page1, page2, page3.
There are eight frames frame0, frame1,………frame7.
According to page table (in next slide) page0, page1,page2,page3 are mapped with frame5, frame6,frame1 and frame2 respectively.

Paging example for a 32 byte memory with 4 byte pages contd.



Thank You